Using BERT to detect transportation mode on raw GPS data

ABSTRACT

Transportation mode detection is a crucial task in the field of people flow pattern recognition by providing insights into travel behaviors and patterns. Inferring transport pattern from global position system(GPS) tracks is especially challenging because GPS data is usually sparse and not absolutely accurate. How to use GPS trajectories to identify users' modes has attracted wide interest over past decade. Various deep learning models have been proposed and widely used in transportation mode identification and have achieved advanced detection accuracy. The recent success of large language models has resulted in state-of-the-art performance in multiple domains, which show its power to capture the message within sequence. In this paper, we use the BERT-based framework to perform transportation mode detection task on GPS trajectories, and use simple motion attributes to implement experiments and compare with classic baseline methods. The experimental results show that BERT-based framework can learn high-level representation of GPS data from the joint sequence of grid and speed, which achieves the best performance with 84.0% accuracy over some classic methods on GeoLife. And the weights pre-trained on large-scaled corpus can surprisingly improve the performance when we encode feature sequences with vocabulary provided by BERT.

1 INTRODUCTION

Transportation mode detection(TMD) plays a pivotal role in transportation planning and personalized travel services. Acknowledging of how the people move at the given time point or predicting the mode in the future is the vital target of the work in TMD field.

Traditionally, travel information was collected through travel surveys[22][19] in the forms of face-to-face interview, mail, etc., which has disadvantages including time-consuming, low response rate and invalid information due to traveler's false memory. With the development of global position system(GPS), many works paid efforts to identify travel behaviour from GPS trajectory data. One of the challenges in TMD is that public labeled GPS data still have limitations in terms of quantity, coverage, and accuracy. It's difficult to detect transportation mode directly from longitude and latitude after training on limited-scaled dataset. The most used dataset GeoLife[26][27][28][25] also can not support models to learn excellent representations directly from raw GPS sequences.

Recent researches mainly focus on how to extract representative features from GPS data and how to improve model. A widely acknowledged idea is that simple motion features(speed, acceleration, jerk) computed directly from GPS trajectories contain dynamic motion messages and help models to learn the representations well. In the feature level, there are abundant works trying to create useful handcrafted features to help machine learning models inferring transportation mode[7][20][2][19][9][16][26][14][24][15][3]. Besides, recent works using deep learning models hopes that models can learn high-level representations from hybrid motion features(relative distance, speed, acceleration, jerk) instead of various handcrafted features[4][23][8] . In the model level, traditional machine learning methods including Random Forest[9], Decision Tree[16][26][15][3], Support Vector Machine[26][24][3], Neural Network[7][2][19][16] have been put into practice. And deep learning frameworks like Convolutional Neural Network[4], Recurrent Neural Network(including LSTM)[8][23], Transformer[11][17] also achieve remarkable results in TMD.

One thing worth discussing is that why multiple motion features should be extracted and then fed into models and can we simplify the input. In [26], length, mean velocity, top three velocity and expectation velocity serves as input variables. And low speed rate, 95% percentage speed, average absolute acceleration are adopted in other work[19]. Doubtfully, why these extracted features can improve the performance? Why should we pick top three velocity instead of top four or five? To eliminate these doubts, in [4] convolutional structured model is utilized to learn high-level representation based on relative distance, speed, acceleration and jerk. And most deep learning based frameworks[23][8] adopt this idea. There are also some other works using different input forms like discrete wavelet transforms[21], ordinal patterns[3], point clouds[10]. But these input forms are also constructed from compound motion attributes including relative distance, speed, acceleration, jerk and so on. Intuitively, can we use only GPS sequence or features as few as possible to infer user's transportation mode from trajectory segments with relative high accuracy?

BERT[5], which stands for Bidirectional Encoder Representations from Transformers, is a typical pre-trained Large Language Model in NLP field. And it has shown its power in NLP, which is recently widely-used in many fields[6][12]. Detailed introduction of BERT can be acquired in section III. It's designed to handle single sequence(up to two sentences) in the downstream tasks including single sentence classification and experimental results show that it can learn the relations of tokens within sequence well to achieve state-of-the-art performance in 11 downstream tasks. The recent success of LLM illustrates us whether it can also be applied in identifying transportation modes, as we can draw an analogy between trajectory segments and language sentences considering each GPS data point as a word in sentence. Especially, the transportation mode detection from GPS trajectory segments is intrinsically a classification task connecting sequence input to label output, which is closely akin to the Single Sentence Classification task tested in [5].

In this paper, we implement a simple transportation mode detection model, which is based on BERT, Without various preprocessing steps including extracting handcrafted features, detecting abnormal movement, etc., we simply feed single sequence(up to two features) to BERT-based model to derive the predictive mode and compare our model to different models with hybrid features on classic annotated dataset GeoLife. Our work aims to reduce abundant and personalized feature extraction steps using fewer simple features to infer transportation mode. And to the best of our knowledge, it's the first time in TMD filed, that LLM framework(BERT in this work) is used to learn high-level representation of GPS trajectory tracks to detect transportation modes, which shows its potential to encode GPS trajectory message.

The main contribution of this paper are summarized as follows:

- To the best of our knowledge, it's the first time that LLM framework(BERT in this work) is used to detect transportation mode from GPS trajectories, which is a groundbreaking work.
- Under the metrics of accuracy and F1-score, BERT-based model can use single sequence concatenating grid sequence and speed sequence to outperform recent advanced models with hybrid motion features.
- We directly use vocabulary dictionary of 'bert-base-uncased' and surprisingly find that pre-trained weights on corpus can help BERT to identify transportation modes more accurately.

The rest of paper is structured as following. In Section II, we review related works in two aspects including feature extraction and model selection. In Section III, we introduce some preliminary knowledge. In Section IV, we describe our BERT-based framework in detail. Experiments are conducted in Section V. And conclusion are drawn in Section VI.

2 RELATED WORK

This section provides an overview of the feature extraction and model selection in TMD tasks.

Table 1: Previous works organized by feature extraction

Lead Author	Features
Zheng, Y.(2008)[26]	Length, mean velocity, expectation of velocity, top three velocity and top three acceleration
Reddy, S.(2010)[15]	Velocity, accelerometer variance, accelerometer DFT components from 1-3 HZ calculated
Gonzalez, P.A.(2010)[7]	Average speed, maximum speed, estimated horizontal accuracy uncertainty, percent Cell-ID fixes, standard deviation of distances between stop locations and average dwell time
Xiao, G.(2015)[19]	Low-speed point rate, travel distance, average speed, average absolute acceleration, median speed, and 95% percentile speed
Dabiri, S.(2020)[4]	Relative distance, velocity, acceleration, and jerk
Nawaz, A.(2020)[13]	Weekday, 4 × Transportation modes probabilities, Interquartile Mean of Velocity and Acceleration
Cardoso-Pereira, P.(2022)[3]	Probability of self-transition, permutation entropy, statistical complexity
Zeng, J.(2023)[23]	Distance interval, time interval, velocity, acceleration, angular velocity, distance to nearest bus stop, and number of bus stops nearby
Löwens, C.(2023)[11]	2D Cartesian coordinates, time gap, velocity

2.1 Feature Extraction

The features proved to be helpful for TMD tasks can be roughly divided into motion features, GIS features, and other features. Motion features, such as distance, velocity, heading rate and so on, are computationally derived from spatial-temporal characteristics(location and timestamp) of GPS tracks. Early studies adopt statistical values(like mean, variance, max, min, percentile, etc.) of these motion features to train the machine learning models to do TMD tasks[7][20][19][9][26][24]. And some works combine these motion features with additional features like magnetometer[2], power spectrum of accelerometer signal[14], transport mode probabilities[13] and accelerometer DFT components[15]. Later deep learning methods use simple motion features including distance, speed, acceleration and jerk to infer transport modes[4][23][8]. GIS features are based on researchers' understanding of geographic information and traffic information . In [16], bus stop location and rail line trajectory are taken into consideration. In [23], the distance of nearest bus stop and the number of bus stops are utilized to help model to distinguish car and bus. There are also some other novel features including discrete wavelet[21], ordinal patterns[3], point clouds[10]. Above is briefly summarized in Table1.

Table 2: Previous works organized by model selection

Model	Works
Random Forest	Stennth, L.(2011)[16], Lari, Z.A.(2015)[9], Cardoso-Pereira, P.(2022)[3]
Decision Tree	Zheng, Y.(2008)[26], Cardoso-Pereira, P.(2022)[3]
Neural Networks	Gonzalez, P.A.(2010)[7], Byon, Y.(2015)[2]
Support Vector Machine	Zheng, Y.(2008)[26], Zhang, L.(2011)[24], Cardoso-Pereira, P.(2022)[3]
Convolutional Neural Network	Dabiri, S.(2020)[4], Tian, A.(2021)[17] combined with transformer, Kim J.(2022)[8] combined with LSTM, Zeng, J.(2023)[23] combined with RCRF
Transformer	Löwens, C.(2023)[11]

2.2 Model Selection

Traditional machine learning methods, like RF[16], DT[26], SVM[3], HMM[15] etc., have been exploited by many researchers. Many different kinds of handcrafted features are constructed to help these models identify the transportation modes. In last decade, deep learning frameworks have made great advance in TMD tasks, such as CNN[4], LSTM[21], Transformer[11]. These methods show better performance in feature extraction and mode recognition. In [4][8][23], CNN-based model is trained to represent motion features effectively in latent space for further classification study. And PointNet is applied to learn representation in [10], which is originally developed in point cloud processing in computer vision. Besides, transformer encoder is used to learn latent embedding for each point within trajectory sequence in [11][17]. Above is briefly summarized in Table2.

3 PRELIMINARIES

In this section, we introduce the preliminaries required to comprehend Transportation Mode Detection tasks and our BERT-based framework.

3.1 GPS Trajectory

A user's GPS Trajectory $\mathbf{T}$ is defined as a sequence of GPS points $\mathbf{p} \in \mathbf{T}$, $\mathbf{T} = (\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_N})$, consisting of $\mathbf{N}$ points. And each GPS point $\mathbf{p}$ is defined as a tuple of latitude, longitude and timestamp, $\mathbf{p} = (\mathbf{lat}, \mathbf{lon}, \mathbf{t})$, which identifies the location of point $\mathbf{p}$ at timestamp $\mathbf{t}$.

3.2 GPS Segment

A GPS segment is defined as a subdivision sequence of a GPS trajectory with the same transportation mode. It is denoted as $\mathbf{S} = (\mathbf{p_1^l}, \mathbf{p_2^l}, \ldots, \mathbf{p_M^l})$, where $\mathbf{p_i^l} \in \mathbf{T}$, $(\mathbf{i} = \mathbf{1}, \ldots, \mathbf{M})$ is annotated with same mode $\mathbf{l} \in \mathbf{L}$, $\mathbf{L}$ is the set of transportation modes in the whole dataset. And $\mathbf{M}$ indicates the number of GPS points within $\mathbf{S}$.

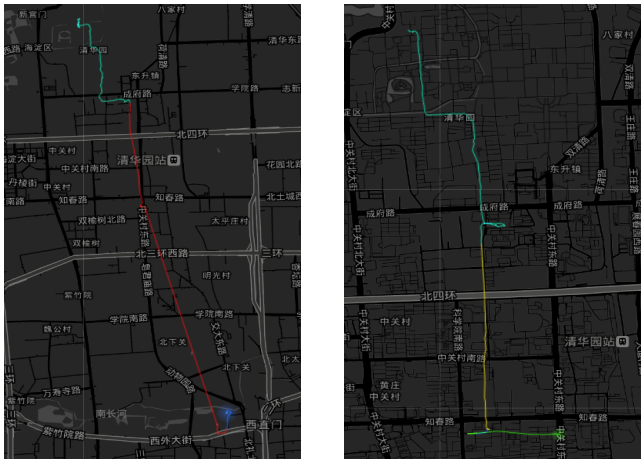

Figure 1: Segments within two individual trajectories annotated with modes. Different color indicates different mode. Blue: Walk, Cyan: Bike, Green: Bus, Yellow: Drive, Red: Train.

3.3 Simple Characteristics of GPS Points

Several simple features are computed for each GPS point $\mathbf{p}$ based on its location and timestamp. First, location of GPS point can be converted into a grid number which will generate a single feature sequence for a GPS segment. We simply set each grid as a square with geographical side length **gird_size**, thereby, mapping all of the GPS points into a big square grid map with **grid_num** grids of one side. Shortly, we set a big square to contain all of the GPS points and divide this big square into smaller squares of side length **grid_size**. And here geographical distance between two GPS points is obtained by Vincenty's formula[18], which is a common and accurate method for computing the geographical distance between two points on the surface of a spheroid. Then grid number $\mathbf{id_i}$ for given GPS point $\mathbf{p_i}$ can be computed using following equation:

$$\begin{aligned} \mathrm{row_i} &= floor(Vincenty(p_i, (lat\_max, p_i[lon]))/grid\_size) \\ \mathrm{col_i} &= floor(Vincenty(p_i, (p_i[lat], lon\_min))/grid\_size) \\ \mathrm{id_i} &= col_i \times grid\_num + row_i \end{aligned} \tag{1}$$

Relative distance $\mathbf{RD_i}$ between two successive GPS points $\mathbf{p_i}$, $\mathbf{p_{i+1}}$ within a GPS segment $\mathbf{S_j}$ can be derived also by Vincenty's formula[18]. Time interval $\mathbf{\Delta t_i}$ between consecutive GPS points can be easily computed by subtracting adjacent timestamps. Then some kinematic fundamental characteristics of each GPS point are available, such as velocity $\mathbf{V_i}$, acceleration $\mathbf{A_i}$ and jerk $\mathbf{J_i}$. Velocity is the rate of change in distance, showing how fast a user is traveling. And acceleration is the rate of change in velocity which indicates the degree of the velocity changes. Jerk is used to describe the rate of change in acceleration. To standardize, these motion feature values are all rounded to **two decimal places**. Given a GPS point $\mathbf{p_i}$, simple motion characteristics for this GPS point can be attained by the following equations:

$$\mathrm{RD_i} = \mathrm{Vincenty}\,(p_i[lat, lon], p_{i+1}[lat, lon]) \tag{2}$$

$$\Delta \mathrm{t_i} = p_{i+1}[t] - p_i[t] \tag{3}$$

$$\mathrm{V_i} = \frac{RD_i}{\Delta t_i} \tag{4}$$

$$\mathrm{A_i} = \frac{V_{i+1} - V_i}{\Delta t_i} \tag{5}$$

$$\mathrm{J_i} = \frac{A_{i+1} - A_i}{\Delta t_i} \tag{6}$$

3.4 Transportation Mode Detection Task

Given training data $\{\mathbf{X_i}, \mathbf{l_i}\}_{i=1}^{n}$ for n samples of $\mathbf{S_i}$ with corresponding mode $\mathbf{l} \in \mathbf{L}$, where $\mathbf{L}$ is the set of transportation modes in the whole dataset, the task is defined as building the optimal classifier to detect transportation mode $\mathbf{l}$ of a user's GPS segment based on the its features $\mathbf{X}$. Here features of a segment consist of **id**, **V**, **A**, **J**, etc.

3.5 BERT

BERT[5] is a method for pre-training natural language processing (NLP) models that was created and published by Google. BERT,based on the transformer architecture,has significantly improved the state-of-the-art performance on a wide range of NLP tasks.

Unlike traditional NLP models that analyze text sequences in one direction (either from left to right or right to left), BERT is designed to consider the context from both directions, hence the term "bidirectional" in its name. This allows the model to understand the context and meaning of a word based on all of its surroundings (left and right of the word).

BERT is pre-trained on a large corpus of text, then fine-tuned for specific tasks. The pre-training phase involves two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM, random words in the sentence are masked and the model tries to predict them based on the context provided by the other non-masked words in the sentence. In NSP, the model learns to predict whether one sentence follows another.

The fine-tuning phase involves training the BERT model on a specific task, such as question answering or sentiment analysis, using an additional output layer.

4 OUR BERT-BASED MODEL

In our work, we introduce the BERT model of 'bert-base-uncased' type which was released by Google. This architecture contains 12 layers of transformers blocks with hidden size 768, 12 self-attention heads, and around 110M trainable parameters. And we add a simple linear full-connection layer after the output of the BERT to do classification tasks. The overview of our BERT-based model is shown in Figure3.

In more details, the last hidden layer which corresponds to the [CLS] token was imported to the final full-connection layer of size 768×5. The input token sequence of BERT in NLP is like:

$$< [CLS], my, dog, is, cute, [SEP], he, likes, play, ##ing, [SEP] >$$

Trying to analogize the TMD tasks with Single Sentence Classification and Sentence Pair Classification tasks conducted in[5], the input sequence should be single feature sequence like that of Single Sentence Classification:

$$< [CLS], [id_1], \ldots, [id_M], [SEP] >$$

or like that of Sentence Pair Classification:

$$< [CLS], [id_1], \ldots, [id_m], [SEP], [V_1], \ldots, [V_n], [SEP] >$$

$[id_i]$ represents the token sequence after id_i is passed through BERT tokenizer. Here tokens in feature sequence are encoded using vocabulary provided by 'bert-base-uncased'. For example, supposing id_1 = 13782772785174, the tokenization step will convert it into

$$137, ##8, ##27, ##7, ##27, ##85, ##17, ##4$$

using its vocabulary. In addition to use BERT loaded with pre-trained weights on its vocabulary(we call it **'preBert'** in our work), we also introduce the same architecture BERT(we call it **'npreBert'**), which was not pre-trained on corpus before. The way to load such a BERT is avoiding to load the weights file and randomly initialize the parameters. Hopefully, we expect these pre-trained weights can improve the performance in TMD, because we adopt the same approach of tokenization and analogize classification tasks in NLP to TMD.

5 EXPERIMENT AND RESULT

5.1 Data Description

The dataset we use is collected in GeoLife[26][27][28] project by 182 users from April 2007 to August 2012. It contains 17,621 trajectories with a total distance of 1,292,951kilometers and a total duration of 50,176 hours. And trajectories are at various sampling rate. In our work we only use the GPS trajectory data annotated with transportation modes, which contains 4,220,336 GPS points, after resampling at 1HZ and selecting the modes. The transportation modes we choose in GeoLife are walk,bike,car,taxi,bus,train,subway, which are the main part and classically used in previous works. 'Drive', in the table below, consists of car and taxi. And 'Train' is composed of train and subway. The number of GPS points annotated with each mode is shown in Table3.

Table 3: Statistics of experimental dataset

Mode	GPS points	GPS Segments
Walk	898751	29490
Bike	712597	22977
Bus	1149888	37419
Drive	673398	22819
Train	785702	26464

5.2 Data Preprocessing

The first step is to normalize the data frequency, which is achieved by resampling. The downsampling of GPS points should preserve the actual points rather than aggregation(e.g., averaging) or estimation(e.g., interpolation)[1]. Therefore, only first record each second remains during the resampling. Then, we remove stopping points within trajectories, which means their positions don't change with time. Then, we split trajectories without records in 10min or longer than 10km into two trajectories and remove those shorter than 1km. Next, The trajectories with less than 10 GPS records are deleted. In addition, we remove the trajectories whose range of movement is within 1km, which is deemed as stay points. After above steps, we preserve the records with modes including walk, bike, bus, car, taxi, train, subway. Furthermore we merge the modes as introduced in Section5.1 to obtain dataset with transportation modes set $\{Walk, Bike, Bus, Drive, Train\}$.

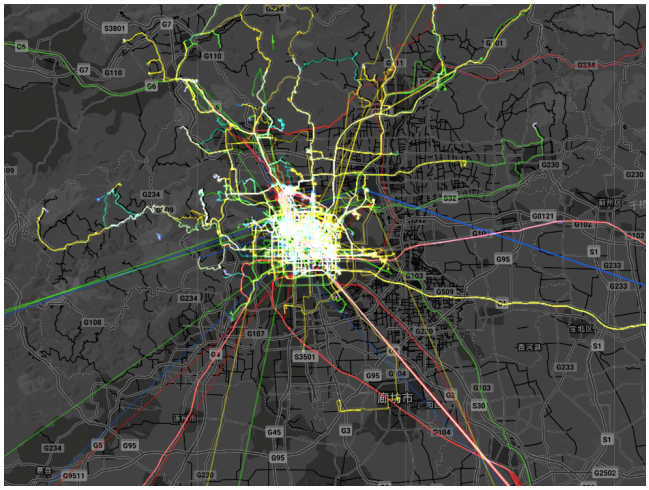

Figure 2: Data around Beijing Region annotated with modes after preprocessing. Blue: Walk, Cyan: Bike, Green: Bus, Yellow: Drive, Red: Train.

5.3 Evaluation Metrics

To evaluate the performance of our model in different aspects, we use two classic metrics in classification tasks.

(1)**Accuracy**: It is computed as the fraction of GPS segments in the test set that are correctly classified.

Accuracy ranges from 0 to 1, where higher score indicates better performance for classification.

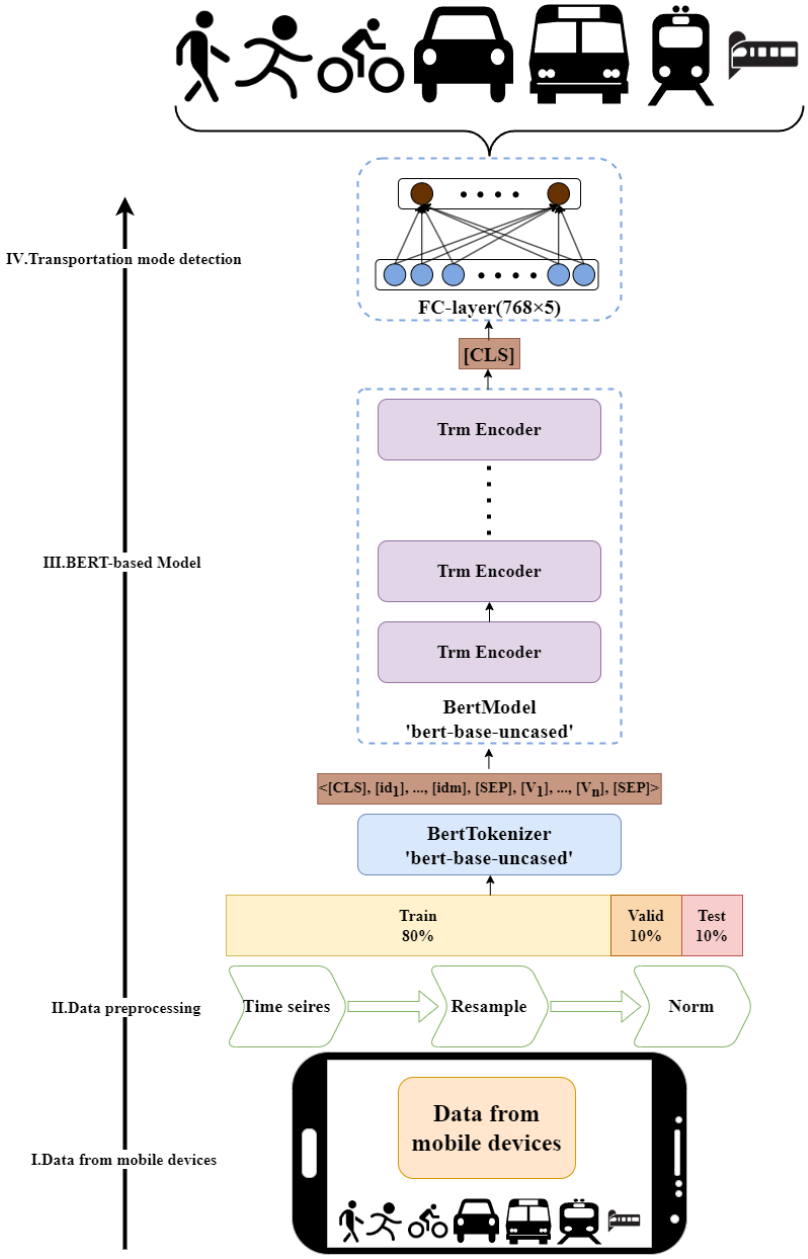

Figure 3: A framework of BERT-based model.

(2)**Weighted F1-score**: F1-score is the harmonic mean of precision and recall:

$$F1-\ \text{score}\ =\frac{2\times\ \text{precision}\ \times\ \text{recall}}{\text{precision}\ +\ \text{recall}} \tag{7}$$

The F1-score ranges from 0 to 1, where 1 indicates perfect precision and recall, and 0 indicates that either the precision or the recall (or both) are zero.

5.4 Methods to Compare

In addition to 'npreBert' used to show the effect of the weights pre-trained on corpus, we adopt several classic machine learning and advanced deep learning methods with public code to compare with our 'preBert'. And following their works, we reproduce these models with best configuration reported in their papers to detect 5 classic transportation modes $\{Walk, Bike, Bus, Drive, Train\}$ as introduced in Section5.2.

(1)**Handcrafted motion features based models**: These machine learning self-supervised models utilize handcrafted motion

features introduced in[26][27] including relative distance, average velocity, expectation velocity, variance of velocity, top three velocity, top three acceleration, heading rate change, stop rate and velocity change rate. K-Nearest Neighbors(KNN), RBF-based Support Vector Machine (SVM), Decision Tree(DT) and Multilayer Perceptron(MLP) are taken into consideration which are implemented in cite[4].

(2)**SECA**[4]: It achieves a deep semi-supervised convolutional autoencoder architecture to use both labeled and unlabeled GPS segments to reach advanced performance. Based on simple GPS characteristic introduced in Section3.3, it leverage convolutional autoencoder to learn high-level representations.

(3)**OPTG**[3]: It creatively introduce Information Theory quantifiers to transportation mode classification. The hybrid features, consisting of probability of self-transition, permutation entropy and statistical complexity, are fed into RBF-based Support Vector Machine and achieves best performance in[3].

(4)**DeepStay**[11]: It is a weakly and self-supervised transformer-based model, focusing on stay regions(SRs) extraction. In Löwens's work, part of experiment is carried on the TMD problems. Based on four features including coordinates, time interval and velocity, it achieves the significantly higher score than some classic baselines[4]. And this work was shortlisted for the Best Paper/Student Paper Award at ITSC 2023.

5.5 Setup

In the experiment, we set the number of GPS points within single GPS segment as 32 (i.e. $\mathbf{M} = 32$, introduced in Section3.2) and thus derive a large number of GPS segments from preprocessed GeoLife. The number of GPS segments within each mode is shown in Table3. Here we choose $\mathbf{M} = 32$ because it's a proper number for containing all the message of input sequence after we convert into token sequence. Then we compute four GPS characteristics on all of the segments following the formulae introduced in Section3.3. When calculating grid number sequence, we tried several values and finally set **grid_size** as **0.1m** to achieve best performance. Considering amount of samples with different transportation modes, we randomly split dataset into **8:1:1** within each class and the merge each part to get train, valid and test dataset. When training the model, early stopping strategy is used to reduce overfitting. And Table4 lists all the parameters we used in training.

Table 4: Parameters in training

	Values	
Parameter	preBert	npreBert
Learning rate	$1e^{-5}$	$1e^{-6}$
Epochs	10	40
Batch size	2(for train and val) , 16(for test)	
Dropout of FC-layer	0.5	
Optimizer	Adam	
Loss function	CrossEntropy	
DataLoader	RandomSampler(for train) SequentialSampler(for val and test)	
Scheduler	get_linear_scheduler_with_warmup	

5.6 Result and Analysis

Table5 and Table7 show all of the experimental results on GeoLife of all the models mentioned above. Table5 records the performance of our BERT-based framework based on different feature input. And Table7 compares our best model with the baseline methods. In the experiment, we evaluate different combination of four features both on 'preBert' and 'npreBert' and compare with baseline methods. The detailed statistics show that BERT loaded with pre-trained weights on corpus, taking joint sequence of grid and velocity as input, achieves the best performance with 84.0% accuracy. Here, DeepStay[11], which demonstrates itself as state-of-the-art, shows unsatisfactory performance. One thing to mention is that it achieves pointwise detection instead of segmentwise, but in[11], accuracy of 83.0% and f1-score of 83.1% are reported to illustrate DeepStay significantly outperforms SECA[4]. Directly using the result shown in [11], our work still realizes improvements both on accuracy and f1-score.

Focusing on feature selection, as shown in Table5, BERT taking compound features usually performs well compared to single feature. It intuitively conforms to previous works as we discussed in Section2. Furthermore, we spot that BERT with hybrid features including grid always achieves relatively high scores. It is reasonable because grid sequence mostly contains location message compared to other three motion characteristics. Due to traveler's habit, there will be many similar GPS segments within one's GPS trajectories. For example, one of the users used to walking to the bus stop, taking the bus for a distance and finally arriving the workspace by foot. These daily activities can generate many similar GPS trajectories with the same subdivided segments. Consequently, grid sequence can help model to learn this relationship within sequence. However, when considering single feature sequence, BERT with velocity performs significantly best. It can be explained intuitively that although grid sequence can help model to capture similar GPS segments owing to travel habit it has poor generalization ability. Because of the data distribution of GeoLife, model perhaps memorizes the characteristics of the trajectories within Beijing and performs poorly on the trajectories within other regions. Velocity, which preserves both spatial and temporal message to some extent, omits the location and focuses on the motion message between GPS points. And other motion features are all extracted from velocity, inevitably losing some important information that the model may learn. Therefore, it's reasonable that BERT with velocity will outperform BERT with other single feature.

And the experimental results of 'preBert' and 'npreBert' show that BERT loaded with pre-trained weights on large corpus always performs better both on accuracy and f1-score. As introduced in Section4, we directly use BertTokenizer with its own vocabulary instead of constructing a new tokenizer and building a new large vocabulary. The inspiration of analogizing TMD tasks to NLP classification tasks make us anticipate that 'preBert' will outperform 'npreBert' as it can utilize what it learned on the same vocabulary. Actually the experimental results conform with our supposition.

To answer the question 'can we use features as few as possible to infer user's transportation mode from trajectory segments with

Table 5: Performance of BERT-based frameworks on different feature input

Feature	preBert		npreBert	
	Accuracy	F1-score	Accuracy	F1-score
I[1]	0.769	0.769	0.664	0.657
V[1]	0.802	0.800	0.757	0.756
A[1]	0.676	0.673	0.446	0.428
J[1]	0.622	0.622	0.445	0.418
I[1]+V[1]	**0.840**	**0.840**	0.775	0.774
I[1]+A[1]	0.803	0.803	0.674	0.672
I[1]+J[1]	0.804	0.803	0.709	0.705
V[1]+A[1]	0.760	0.759	0.714	0.713
V[1]+J[1]	0.764	0.763	0.726	0.721
A[1]+J[1]	0.635	0.634	0.501	0.493

[1] I:Id, V:Velocity, A:Acceleration, J:Jerk

relative high accuracy?' mentioned in Section1, we compare our BERT-based model with best performance to the baselines ranging from classic machine learning methods to recent deep learning methods. And as shown in Table7, our best model achieves significant improvements both on accuracy and f1-score which taking joint sequence of grid and velocity as input.

Table6 shows the confusion matrix of our BERT-based model with features I+V. Although the train class does not constitute the largest portion of the dataset in Table3, it seems our best model can learn excellent representation of train segments with selected features and accurately identify with both high precision and recall. Meanwhile it performs worst on the drive mode. One possible reason is that the drive mode takes the smallest portion of the dataset. Besides, bus mode and drive mode are difficult to tell apart which is consistent with the conclusion in previous works[4][23].

Table 6: Confusion Matrix for our BERT-based Model with features I+V

	Predicted Modes					
True Modes	**Walk**	**Bike**	**Bus**	**Drive**	**Train**	**Recall**
Walk	2429	126	200	62	50	0.847
Bike	134	1980	114	24	6	0.877
Bus	254	142	2878	324	54	0.788
Drive	51	28	383	1675	50	0.766
Train	41	2	60	52	2392	0.939
Precision	0.835	0.869	0.792	0.784	0.937	-

6 CONCLUSION

In this paper, we draw an analogy between the transportation mode detection task and the classification tasks in NLP, trying to use LLM framework to capture message within GPS segments. Using fewer features compared to previous works, our BERT-based model takes single sequence as input. And by using its own tokenizer and vocabulary, BERT loaded with pre-trained weights achieves great improvements. Our work gives an illumination about using powerful LLM frameworks to do TMD tasks and leaves room for discussion about pre-training.

In future work, we will discuss more about the pre-training effect of LLM frameworks on TMD tasks and try to implement GPS data based pre-trained framework to improve the performance.

Table 7: Performance comparison between our best model and baselines on GeoLife

Model	Feature	Accuracy	F1-score
preBert	I[1]+V[1]	**0.840**	**0.840**
OPTG[3]	p_{st}[4]+$H_S[p_\pi]$[4]+$C_{JS}[p_\pi]$[4]	0.728	0.683
DeepStay[11]	U[5]+T[5]+V[1]	0.594	0.641
SECA[4]	RD[1]+V[1]+A[1]+J[1]	0.731	0.726
Semi-Pseudo-Label[4]	RD[1]+V[1]+A[1]+J[1]	0.725	0.717
Semi-Layer-Wise[4]	RD[1]+V[1]+A[1]+J[1]	0.663	0.647
Semi-Two-Steps[4]	RD[1]+V[1]+A[1]+J[1]	0.556	0.533
CNN[4]	RD[1]+V[1]+A[1]+J[1]	0.737	0.729
RF[4]	RD[1]+AV[2] +EV[2]+VV[2]+Top3V[3] +Top3A[3]+HRC[2]+SR[2]+VCR[2]	0.780	0.776
KNN[4]	RD[1]+AV[2] +EV[2]+VV[2]+Top3V[3] +Top3A[3]+HRC[2]+SR[2]+VCR[2]	0.579	0.564
SVC[4]	RD[1]+AV[2] +EV[2]+VV[2]+Top3V[3] +Top3A[3]+HRC[2]+SR[2]+VCR[2]	0.454	0.386
DT[4]	RD[1]+AV[2] +EV[2]+VV[2]+Top3V[3] +Top3A[3]+HRC[2]+SR[2]+VCR[2]	0.694	0.695
MLP[4]	RD[1]+AV[2] +EV[2]+VV[2]+Top3V[3] +Top3A[3]+HRC[2]+SR[2]+VCR[2]	0.480	0.431

[1] I:Id, RD:Relative Distance, V:Velocity, A:Acceleration, J:Jerk
[2] AV:Average Velocity, EV:Expectation Velocity, VV:Variance of Velocity, HRC:Heading Rate Change, SR:Stop Rate, VCR:Velocity Change Rate
[3] Top3V:Max Velocity 1,Max Velocity 2,Max Velocity 3; Top3A: Max Acceleration 1, Max Acceleration 2, Max Acceleration 3
[4] p_{st}:Probability of Self-Transition, $H_S[p_\pi]$:Permutation Entropy, $C_{JS}[p_\pi]$:Statistical Complexity
[5] : U: UTM coordinate, T: Time gap

REFERENCES

[1] O. Burkhard, H. Becker, R. Weibel, and K.W. Axhausen. 2020. On the requirements on spatial accuracy and sampling rate for transport mode detection in view of

a shift to passive signalling data. *Transportation Research Part C: Emerging Technologies* 114 (2020), 99–117. https://doi.org/10.1016/j.trc.2020.01.021

[2] Young-Ji Byon and Steve Liang. 2014. Real-Time Transportation Mode Detection Using Smartphones and Artificial Neural Networks: Performance Comparisons Between Smartphones and Conventional Global Positioning System Sensors. *Journal of Intelligent Transportation Systems* 18, 3 (2014), 264–272. https://doi.org/10.1080/15472450.2013.824762 arXiv:https://doi.org/10.1080/15472450.2013.824762

[3] Isadora Cardoso-Pereira, João B. Borges, Pedro H. Barros, Antonio F. Loureiro, Osvaldo A. Rosso, and Heitor S. Ramos. 2022. Leveraging the self-transition probability of ordinal patterns transition network for transportation mode identification based on GPS data. *Nonlinear Dynamics* 107, 1 (01 Jan 2022), 889–908. https://doi.org/10.1007/s11071-021-07059-x

[4] Sina Dabiri, Chang-Tien Lu, Kevin P. Heaslip, and Chandan K. Reddy. 2020. Semi-Supervised Deep Learning Approach for Transportation Mode Identification Using GPS Trajectory Data. *IEEE Transactions on Knowledge and Data Engineering* 32 (2020), 1010–1023. https://api.semanticscholar.org/CorpusID:85524931

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[6] Ifigenia Drosouli, Athanasios Voulodimos, Paris Mastorocostas, Georgios Miaoulis, and Djamchid Ghazanfarpour. 2023. TMD-BERT: A Transformer-Based Model for Transportation Mode Detection. *Electronics* 12, 3 (2023). https://doi.org/10.3390/electronics12030581

[7] P.A. Gonzalez. 2010. Automating mode detection for travel behaviour analysis by using global positioning systems-enabled mobile phones and neural networks. *IET Intelligent Transport Systems* 4 (March 2010), 37–49(12). Issue 1. https://digital-library.theiet.org/content/journals/10.1049/iet-its.2009.0029

[8] Jinsoo Kim, Jae Hun Kim, and Gunwoo Lee. 2022. GPS data-based mobility mode inference model using long-term recurrent convolutional networks. *Transportation Research Part C: Emerging Technologies* 135 (2022), 103523. https://doi.org/10.1016/j.trc.2021.103523

[9] Zahra Lari and Amir Golroo. 2015. Automated Transportation Mode Detection Using Smart Phone Applications via Machine Learning: Case Study Mega City of Tehran. https://api.semanticscholar.org/CorpusID:107496907

[10] Rongsong Li, Zi Yang, Xin Pei, Yun Yue, Shaocheng Jia, Chunyang Han, and Zhengbing He. 2023. A novel one-stage approach for pointwise transportation mode identification inspired by point cloud processing. *Transportation Research Part C: Emerging Technologies* 152 (2023), 104127. https://doi.org/10.1016/j.trc.2023.104127

[11] Christian Löwens, Daniela Thyssens, Emma Andersson, Christina Jenkins, and Lars Schmidt-Thieme. 2023. DeepStay: Stay Region Extraction from Location Trajectories using Weak Supervision. arXiv:2306.06068 [cs.CV]

[12] Mashaal Musleh. 2022. Towards a unified deep model for trajectory analysis. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (<conf-loc>, <city>Seattle</city>, <state>Washington</state>, </conf-loc>) *(SIGSPATIAL '22)*. Association for Computing Machinery, New York, NY, USA, Article 109, 2 pages. https://doi.org/10.1145/3557915.3565529

[13] Asif Nawaz, Huang Zhiqiu, Wang Senzhang, Yasir Hussain, Amara Naseer, Muhammad Izhar, and Zaheer Khan. 2020. Mode Inference using enhanced Segmentation and Pre-processing on raw Global Positioning System data. *Measurement and Control* 53, 7-8 (2020), 1144–1158. https://doi.org/10.1177/0020294020918324 arXiv:https://doi.org/10.1177/0020294020918324

[14] Philippe Nitsche, Peter Widhalm, Simon Breuss, Norbert Brändle, and Peter Maurer. 2014. Supporting large-scale travel surveys with smartphones – A practical approach. *Transportation Research Part C: Emerging Technologies* 43 (2014), 212–221. https://doi.org/10.1016/j.trc.2013.11.005 Special Issue with Selected Papers from Transport Research Arena.

[15] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.* 6, 2, Article 13 (mar 2010), 27 pages. https://doi.org/10.1145/1689239.1689243

[16] Leon Stenneth, Ouri Wolfson, Philip S. Yu, and Bo Xu. 2011. Transportation mode detection using mobile phones and GIS information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (Chicago, Illinois) *(GIS '11)*. Association for Computing Machinery, New York, NY, USA, 54–63. https://doi.org/10.1145/2093973.2093982

[17] Aosheng Tian, Ye Zhang, Huiling Chen, Chao Ma, and Shilin Zhou. 2021. An Ensemble of ConvTransformer Networks for the Sussex-Huawei Locomotion-Transportation (SHL) Recognition Challenge. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers* (Virtual, USA) *(UbiComp/ISWC '21 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 408–411. https://doi.org/10.1145/3460418.3479383

[18] T. Vincenty. 1975. DIRECT AND INVERSE SOLUTIONS OF GEODESICS ON THE ELLIPSOID WITH APPLICATION OF NESTED EQUATIONS. *Survey Review* 23, 176 (1975), 88–93. https://doi.org/10.1179/sre.1975.23.176.88 arXiv:https://doi.org/10.1179/sre.1975.23.176.88

[19] Guangnian Xiao, Zhicai Juan, and Chunqin Zhang. 2015. Travel mode detection based on GPS track data and Bayesian networks. *Computers, Environment and Urban Systems* 54 (2015), 14–22. https://doi.org/10.1016/j.compenvurbsys.2015.05.005

[20] Fei Yang, Zhenxing Yao, and Peter J. Jin. 2015. GPS and Acceleration Data in Multimode Trip Data Recognition Based on Wavelet Transform Modulus Maximum Algorithm. *Transportation Research Record* 2526, 1 (2015), 90–98. https://doi.org/10.3141/2526-10 arXiv:https://doi.org/10.3141/2526-10

[21] James J. Q. Yu. 2021. Travel Mode Identification With GPS Trajectories Using Wavelet Transform and Deep Learning. *IEEE Transactions on Intelligent Transportation Systems* 22, 2 (2021), 1093–1103. https://doi.org/10.1109/TITS.2019.2962741

[22] Yang Yue, Tian Lan, Anthony G.O. Yeh, and Qing-Quan Li. 2014. Zooming into individuals to understand the collective: A review of trajectory-based travel behaviour studies. *Travel Behaviour and Society* 1, 2 (2014), 69–78. https://doi.org/10.1016/j.tbs.2013.12.002

[23] Jiaqi Zeng, Yi Yu, Yong Chen, Di Yang, Lei Zhang, and Dianhai Wang. 2023. Trajectory-as-a-Sequence: A novel travel mode identification framework. *Transportation Research Part C: Emerging Technologies* 146 (2023), 103957. https://doi.org/10.1016/j.trc.2022.103957

[24] Lijuan Zhang, Sagi Dalyot, Daniel Eggert, and Monika Sester. 2012. Multi-stage approach to travel-mode segmentation and classification of gps traces. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 3825 (2012), 87–93. https://api.semanticscholar.org/CorpusID:2594383

[25] Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li. 2011. *Geolife GPS trajectory dataset - User Guide* (geolife gps trajectories 1.1 ed.). https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/

[26] Yu Zheng, Xing Xie, and Wei-Ying Ma. 2008. Understanding Mobility Based on GPS Data. In *Proceedings of the 10th ACM conference on Ubiquitous Computing (Ubicomp 2008)* (proceedings of the 10th acm conference on ubiquitous computing (ubicomp 2008) ed.). https://www.microsoft.com/en-us/research/publication/understanding-mobility-based-on-gps-data/

[27] Yu Zheng, Xing Xie, and Wei-Ying Ma. 2009. Mining Interesting Locations and Travel Sequences From GPS Trajectories. In *Proceedings of International conference on World Wide Web 2009* (proceedings of international conference on world wide web 2009 ed.). https://www.microsoft.com/en-us/research/publication/mining-interesting-locations-and-travel-sequences-from-gps-trajectories/ WWW 2009.

[28] Yu Zheng, Xing Xie, and Wei-Ying Ma. 2010. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Eng. Bull.* 33 (2010), 32–39. https://api.semanticscholar.org/CorpusID:3219429